Exploring Optimal Service Compositions in Highly Heterogeneous and Dynamic Service-Based Systems

Dionysios Efstathiou, Peter Mcburney, Steffen Zschaler, Johann Bourcier

Exploring Optimal Service Compositions in Highly Heterogeneous and Dynamic Service-Based Systems

Dionysios Efstathiou[1], Peter McBurney[1], Steffen Zschaler[1], and Johann Bourcier[2]

[1] Department of Informatics, King's College London
{dionysios.efstathiou,peter.mcburney}@kcl.ac.uk, szschaler@acm.org
[2] IRISA, University of Rennes 1
johann.bourcier@irisa.fr

Abstract. Dynamic and heterogeneous service-oriented systems present challenges when developing composite applications that exhibit specified quality properties. Resource heterogeneity, mobility, and a large number of spatially distributed service providers complicate the process of composing complex applications with specified QoS requirements. This PhD project aims at enabling the efficient run-time generation of service compositions that share functionality, but differ in their trade-offs between multiple competing and conflicting quality objectives such as application response time, availability and consumption of resources. In this paper we present a research roadmap towards an approach for flexible service composition in dynamic and heterogeneous environments.

1 Introduction

Pervasive and mobile computation is about systems consisting of a large number of computational nodes with heterogeneous capabilities communicating over highly dynamic networks. *Service composition* [1] is an appropriate paradigm for creating complex applications in such environments where nodes' resources, such as data, network and hardware components, are offered as software services.

Mobile computing and pervasive applications present unique challenges when trying to compose applications with specified QoS due to the resource heterogeneity and high dynamism of both nodes and underlying networks caused by their inherent mobility. When composing services in such systems, continuous adaptation and optimisation must maintain the required functional and QoS levels of the application as the system evolves. The motivating scenario of our research is a fire-fighter decision support system [2]. The goal of the system is to combine services provided by heterogeneous devices such as sensors and tablets, to compose complex applications for assisting fire-fighters to make well-informed decisions within a crisis. For example, during a forest fire, a commanding officer may combine information (e.g. fire position, weather conditions) and prediction services for estimating the evolution of the fire. The composed application must exhibit minimum response time to respect the above time-critical scenario.

Traditional approaches try to optimise the QoS of a composite application by considering only the selection of which concrete services to be coordinated by a central orchestrator [3, 4]. However, they neglect to consider how the coordination of a composite application may affect its quality. To address this issue, we propose a flexible service composition model which considers the following three *degrees of freedom* (DoFs) for modifying the quality of a composition: (a) service selection, (b) orchestration partitioning, and (c) orchestrator node selection. These DoFs formulate the space of candidate configurations all of which realise the functionality of the targeted composite application, but each of which exhibits different QoS properties. The highly dynamic nature of the studied systems require timely exploration of configurations which best satisfy user's goals. This PhD aims to provide a run-time optimisation-based approach for automatic exploration of trade-off composite applications that share functionality, but where each of them differs in their quality trade-offs.

The paper is organised as follows. Section 2 presents the background to our research followed by Section 3 which states our research problem and roadmap. Section 4 describes the proposed approach. Section 5 discusses related work, before Section 6 summarises the expected contributions of this PhD.

2 Background

Service composition creates complex applications by aggregating services to provide composite functionality that none of the services could provide by itself. We use the concepts of *Concrete Service* (CS) which refers to an invocable service, and *Abstract Service* (AS) which abstractly defines the functionality of a service. Let a composite application be represented as a directed graph, as shown in Fig. 1, consisting of a node set $AS = \langle AS_1, AS_1, \cdots, AS_n \rangle$ of abstract services and an edge set $DF = \langle (AS_i, AS_j) : \ i \neq j, \ 1 \leq i \leq n, 1 \leq j \leq n \rangle$ of data flow between abstract services, where AS_i is the source and AS_j the data destination.

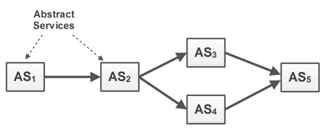

Fig. 1: An example abstract workflow plan

Currently, service composition is identified with *Orchestration* where a central entity coordinates the control and data flow between participating services. *Choreography* defines the interaction protocol between services from a global perspective with an emphasis on P2P collaboration. *Decentralised Orchestration* [5] lies between these two extremes where the coordination of the application is distributed to many nodes. Each orchestrating node integrates a local workflow engine and has only a partial view of the overall composition. The orchestrators cooperate with one another towards realising the complete application.

3 Problem Statement and our Research Roadmap

In systems consisting of a large number of highly dynamic and heterogeneous nodes, the approaches of centralised orchestration and fully decentralised choreography result in suboptimal configurations [6] as they do not exploit the dynamic resource heterogeneity of both the nodes (e.g. battery) and the underlying networks (e.g. topology, bandwidth).

To fill this gap, we propose the research plan depicted in Fig. 2 where darker colours indicate higher progress achieved so far. The first goal of this PhD project is to select an appropriate application scenario. The chosen scenario is a firefighter decision support system where heterogeneous nodes (i.e. sensors, tablets) are deployed in an area of interest and form an ad-hoc service-based network [2]. Then, we focus on providing a service composition model by enabling flexible coordination of composite applications. In step 3, based on the chosen model, we choose an appropriate representation of the space of possible composition configurations. Step 4 concerns the identification of the metrics of interest which will be used as the optimisation objectives towards exploring optimal composition configurations. To estimate the real objective function of a composition configuration we use a resource-consuming simulation. As search algorithms require a large number of objective function evaluations to converge to a set of optimal solutions, we propose the replacement of the expensive simulation by surrogate models for approximating the quality values of a composition configuration (step 5). After choosing the problem representation and an appropriate (surrogate) objective function for guiding the search process, the next step is to define a suitable search algorithm for exploring trade-off composition configurations which share functionality but exhibit different QoS trade-offs. We will particularly investigate the applicability of stochastic metaheuristics. Finally, we plan to validate the applicability of the overall approach and use the finding to refine the chosen representation, simulation, and optimisation.

4 The Proposed Approach

We now present our approach to enable automated exploration of optimal composite applications in dynamic and heterogeneous service-based environments.

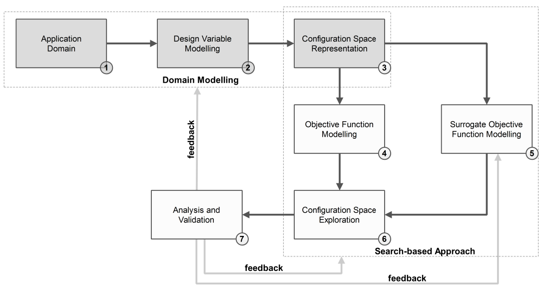

Fig. 2: The research roadmap of this PhD

4.1 Modelling Flexible Service Composition

Firstly, we propose a formulation for flexible composition of applications in dynamic and heterogeneous systems. We call the *Degree of Freedom* (DoF) a parameter of a configuration which is free to be varied to affect application's QoS while leaving its functional part unaffected. We propose the consideration of the following three DoFs for modifying a composition configuration: (a) selection of particular concrete services to implement the abstract services of the composition plan; (b) distribution of the control of the overall application into sub-orchestrations; and (c) selection of the nodes to realise the various sub-orchestrations. The set of choices for realising a composite application is called *service composition configuration*, while the set of all possible configurations is called *design space*.

Current composition approaches try to optimise the quality of a composite application by considering only the selection of particular services to participate in a centralised orchestration. However, in the context of highly mobile resource-constrained systems which are characterised by intermittent connectivity, it is hard to assume that a resource-rich orchestrator has a reliable access to all services in a composition or that such an entity even exists. The proposed approach chooses flexibly the appropriate level of decentralisation for a composite application based on the resource availability of the participating nodes.

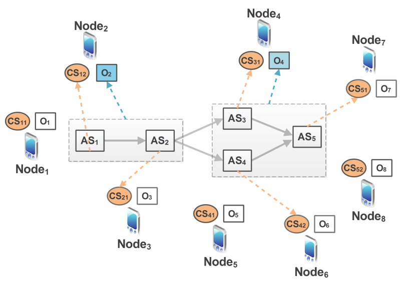

Fig. 3: An example concrete service composition configuration

In Fig. 3, 8 nodes share their services for composing the application in Fig. 1. To define a composition configuration, we have to make decisions for each of the three mentioned DoFs. Firstly, we have to select which concrete services to implement the abstract services of the application (1st DoF). In our example, CS_{12} is chosen for realising AS_1 out of two possible choices (CS_{11} and CS_{12}). Then, the initial composite application is decomposed into two sub-orchestrations (2nd DoF). Finally, Node_2 is responsible for coordinating the first sub-orchestration and Node_4 for the second one (3rd DoF).

4.2 Formulating our Optimisation Problem

When considering multiple criteria, such as response time and battery consumption, there is no single optimal solution. Instead, there exist a set of Pareto-optimal solutions. Without having further information about user's goals, none of these trade-off solutions can be said to be better than another. To enable the exploration of trade-off configurations, we define the following:

Definition 1 Distributed Service Orchestration Problem
Given: A set of m abstract services that compose the service composition plan P, a set of n nodes where each node provides a single concrete service and can coordinate a single orchestration, a mapping between concrete service implementations and abstract services, and a set of q quality objectives $Q = \{Q_1, \cdots, Q_q\}$.
Problem: Find a representative set of composition configurations, all of which implement the functionality described by P, but differ in their quality trade-offs according to Q, by using different design options for the mentioned DoFs.

To solve an optimisation problem we have to define two key ingredients [7]: (i) a problem representation, and (ii) an objective function. The last step is to employ an optimisation algorithm for exploring the solution space.

Representation We have designed a metamodel for specifying the space of possible composition configurations that realise a composite application, which is omitted due to lack of space. Based on this metamodel, we are able to produce model instances like the one described in Fig. 3.

Objective Function To simulate the studied scenario, we plan to use the NS-3[3] to measure the QoS properties (or else optimisation objectives) of candidate configurations. Search-based algorithms require a large number of objective function evaluations to explore a set of trade-off solutions. However, the high computational cost of the simulator makes impossible its usage for runtime exploration of trade-off configurations.

Surrogate-Based Optimisation (SBO) [8] aims at reducing the computational time of optimisation problems by replacing expensive objective functions with surrogate models. The goal of surrogate models is to approximate the values of the real objective functions as close as possible and also be orders of magnitude cheaper to compute. Surrogate models can be built by using data samples produced by simulation runs for predicting quality properties of composition configurations, such as response time and battery consumption.

4.3 Validation Strategy

The goal is to understand the ability of our approach to explore promising composition configurations by various simulation-based empirical studies. The use of simulation avoids the costly and impractical physical testing of such systems and gives us statistical power by allowing repeated experiments. Firstly, we aim to study the suitability of the proposed formulation for composing applications in highly dynamic and heterogeneous environments. Then, we will investigate the ability of the developed surrogate models to guide the search towards promising areas of the search space. Finally, we plan to study the ability of the developed SBO technique to provide configurations of high-quality during runtime by measuring the execution time for exploring solutions of acceptable quality.

[3] `http://www.nsnam.org/`

5 Related Work

QoS-aware service composition approaches can be grouped into centralised and decentralised. In the first category, the goal is to find the set of concrete services to participate in a centralised orchestration that offer the required functionality, respect user's preferences and constraints, and optimise composition's QoS [3, 4]. However, these approaches focus on networks with abundant bandwidth and stationary nodes and neglect to study the problems of centralised orchestration.

On the other hand, Schuhmann *et al.* [6] proposed a hybrid configuration of distributed applications in the context of pervasive environments by adjusting the level of decentralisation based on the number of available resource-rich nodes. Fdhila *et al.* [9] proposed a decentralised approach for composing applications by decentralising a composition into partitions of services that communicate frequently, towards optimising the overall QoS of the composition. However, existing approaches do not adapt automatically to changing conditions such as resource availability and network connectivity.

6 Contributions

The expected contributions of this PhD can be summarised as: (i) a design space formulation for flexible composition of applications in highly dynamic and heterogeneous service-based environments; (ii) a search-based approach for efficiently exploring trade-off compositions ; and (iii) a surrogate-assisted approach for accelerating the search process of optimal configurations.

Acknowledgements. This work has been supported by the European FP7 Marie Curie Initial Training Network "RELATE" (Grant Agreement No. 264840).

References

1. Papazoglou, M., Georgakopoulos, D.: Introduction: Service-Oriented Computing. Comm. of ACM **46** (2003) 24–28
2. Efstathiou, D., McBurney, P., Plouzeau, N., Zschaler, S.: Improving the Quality of Distributed Composite Service Applications. In: ICCSW. (2012)
3. Canfora, G., Di Penta, M., Esposito, R., Villani, M.L.: An Approach for QoS-Aware Service Composition Based on Genetic Algorithms. In: GECCO. (2005)
4. Cardellini, V., Casalicchio, E., Grassi, V., Iannucci, S., Presti, F.L., Mirandola, R.: MOSES: A Framework for QoS Driven Runtime Adaptation of Service-Oriented Systems. TSE **99** (2011)
5. Sheng, Q.Z., Benatallah, B., Dumas, M., yan Mak, E.O.: SELF-SERV: A Platform for Rapid Composition of Web Services in a Peer-to-Peer Environment (2002)
6. Schuhmann, S., Herrmann, K., Rothermel, K.: Efficient Resource-Aware Hybrid Configuration of Distributed Pervasive Applications. In: the 8th PerCom. (2010)
7. Harman, M., McMinn, P., Souza, J., Yoo, S.: Search Based Software Engineering: Techniques, Taxonomy, Tutorial. In: Empirical Software Engineering and Verification. Volume 7007 of Lecture Notes in Computer Science. (2012) 1–59
8. Jin, Y.: Surrogate-assisted evolutionary computation: Recent advances and future challenges. Swarm and Evolutionary Computation **1**(2) (2011) 61–70
9. Fdhila, W., Dumas, M., Godart, C., Garca-Bauelos, L.: Heuristics for Composite Web Service Decentralization. SoSyM (2012) 1–21